INSTRUCCIONES PARA EL AGENTE - SISTEMAS DE BOT DISCORD

© CONTEXTO GENERAL

El usuario ya tiene un bot de Discord funcional y quiere integrar **DOS SISTEMAS ESPECÍFICOS** que ya han sido desarrollados como código base. Tu trabajo es adaptar, integrar y completar estos sistemas según las necesidades del bot existente.

🔧 SISTEMA 1: TIEMPO, FECHA, PERIODO DEL DÍA Y CLIMA

OBJETIVO PRINCIPAL

- Crear un sistema que detecte la hora local del usuario según su país
- Generar lógica dinámica del juego basada en tiempo real
- Mostrar información temporal en embeds automáticos y actualizables
- Implementar clima global que afecte al gameplay

COMPONENTES DESARROLLADOS

TimeWeatherSystem (Clase Principal)

TimeWeatherHandler (Manejador)

```
javascript

// Funcionalidades incluidas:
- handleTimeCommand(): Comando principal
- handleCountrySelection(): Procesa selección de país
- updateTimeEmbed(): Actualiza embed automáticamente
```

📕 MODIFICACIÓN REQUERIDA - SELECCIÓN DE PAÍSES

IMPORTANTE: El usuario especifica que la selección de países NO debe ser un menú desplegable independiente, sino **integrada en el sistema de creación de personaje como categorías**.

TU TAREA:

- 1. **Eliminar** el método (createCountrySelectMenu())
- 2. Crear un sistema de categorías de países para el tutorial de creación de personaje
- 3. **Agrupar países** por regiones geográficas (Europa, América, Asia, etc.)
- 4. Integrar la selección en el flujo de creación del personaje

🙀 EFECTOS DEL CLIMA EN GAMEPLAY

```
javascript
```

```
// Efectos ya programados:
    Soleado: +20% loot, enemigos normales
    Lluvia: +10% loot, -20% enemigos
    Niebla: -10% loot, +30% enemigos
    Tormenta: +30% loot, +50% enemigos
    Nevada: +10% loot, -30% enemigos
    Nublado: Sin efectos
```

SISTEMA 2: EXPLORACIÓN CON EVENTOS DINÁMICOS

OBJETIVO PRINCIPAL

- Sistema de exploración continua tipo RPG
- Un solo embed que se actualiza (sin spam de mensajes)
- Eventos aleatorios con diferentes probabilidades
- Navegación entre eventos anteriores

COMPONENTES DESARROLLADOS

ExplorationSystem (Clase Principal)

ExplorationHandler (Manejador)

javascript

```
javascript

// Funcionalidades incluidas:
    handleExplorationStart(): Inicia exploración
    handleExploreAction(): Procesa exploración (2-4 seg delay)
    handleExploreBack(): Navegar evento anterior
    explorationHistory: Historial por usuario (10 eventos max)
    explorationCooldowns: Cooldown de 3 segundos
    processEventReward(): Para integrar recompensas
```

M INTEGRACIÓN CON SISTEMA EXISTENTE

El método (processEventReward()) está preparado para que añadas:

- Añadir ítems al inventario del usuario
- Iniciar sistema de combate
- Dar experiencia/monedas
- Activar quirks especiales
- Cualquier otra lógica del juego

⊘ INSTRUCCIONES DE INTEGRACIÓN

1. ESTRUCTURA DE ARCHIVOS RECOMENDADA

```
/systems/

— time-weather.js (Sistema 1)

— exploration.js (Sistema 2)

— character-creation.js (Modificar para países)

/handlers/

— time-weather-handler.js

— exploration-handler.js

/data/

— user-data.js (Donde guardas userData)
```

2. INICIALIZACIÓN EN EL BOT PRINCIPAL

```
javascript
// En tu archivo principal del bot
const { TimeWeatherSystem, TimeWeatherHandler } = require('./systems/time-weather');
const { ExplorationSystem, ExplorationHandler } = require('./systems/exploration');

// Inicializar sistemas
const timeWeather = new TimeWeatherSystem();
const exploration = new ExplorationSystem();

// Inicializar handlers
const timeHandler = new TimeWeatherHandler(timeWeather);
const exploreHandler = new ExplorationHandler(exploration, timeWeather);

// Conectar con eventos de Discord
client.on('interactionCreate', async (interaction) => {
    // Manejar comandos y botones según customId
});
```

3. ESTRUCTURA DE DATOS DE USUARIO REQUERIDA

4. COMANDOS A CREAR/MODIFICAR

Comando de Tiempo/Clima

```
javascript

// Ejemplo de comando slash
{
    name: 'tiempo',
    description: 'Ver información de tiempo y clima actual'
}
```

Comando de Exploración

```
javascript

// Ejemplo de comando slash
{
    name: 'explorar',
    description: 'Iniciar exploración del mundo'
}
```

IDs de Interacciones a Manejar

© TAREAS ESPECÍFICAS PARA EL AGENTE

PRIORIDAD ALTA - MODIFICACIONES REQUERIDAS

- 1. SISTEMA DE PAÍSES EN CREACIÓN DE PERSONAJE
 - Eliminar (createCountrySelectMenu()) del código
 - Crear categorías geográficas de países
 - Integrar selección en tutorial de creación
 - Mantener el (timezoneMap) intacto

2. P INTEGRACIÓN CON BOT EXISTENTE

- Adaptar la estructura de (userData)
- Conectar con sistema de inventario existente
- Integrar con sistema de combate existente

Añadir manejo de recompensas en processEventReward()

3. **OPTIMIZACIONES Y MEJORAS**

- Añadir sistema de auto-actualización de embeds
- Implementar persistencia de datos
- Añadir validaciones de errores
- Optimizar rendimiento para múltiples usuarios

PRIORIDAD MEDIA - FUNCIONALIDADES ADICIONALES

4. II SISTEMA DE ESTADÍSTICAS

- Contador de exploraciones por usuario
- Estadísticas de eventos encontrados
- Historial de clima y efectos

5. M EFECTOS AVANZADOS DEL CLIMA

- Eventos especiales según clima
- Modificadores temporales
- Interacción con otros sistemas del bot

PRIORIDAD BAJA - PULIR DETALLES

6. PMEJORAS VISUALES

- Embeds más atractivos
- Mejores emojis y formato
- Animaciones de texto

7. **CONFIGURACIÓN AVANZADA**

- Cooldowns configurables
- Probabilidades ajustables
- Nuevos tipos de eventos

NOTAS IMPORTANTES

🔽 LO QUE YA ESTÁ HECHO

- Z Lógica completa de tiempo y clima
- Sistema de exploración funcional
- Z Embeds dinámicos que se actualizan
- Botones interactivos

- Historial de eventos
- Z Efectos del clima en gameplay
- Cooldowns y validaciones básicas

LO QUE NECESITA ADAPTACIÓN

- Selección de países (cambiar a categorías)
- Integración con userData existente
- A Conexión con inventario/combate
- A Comandos slash específicos del bot
- Persistencia de datos

NOTITION NOTITION NOTITION

- S La lógica de zonas horarias (timezoneMap)
- S Los efectos del clima (getWeatherEffects())
- Substitution
 La estructura de eventos de exploración
- S El sistema de embeds dinámicos

EJEMPLO DE FLUJO COMPLETO

Flujo del Usuario:

- 1. Creación de Personaje → Selecciona país por categorías
- 2. **Comando /tiempo** → Ve su hora local y clima global
- 3. **Comando /explorar** → Inicia exploración
- 5. **Efectos del clima** → Modifican probabilidades
- 6. **Recompensas** → Se añaden al inventario/stats
- 7. **Navegación** → Puede ver eventos anteriores

Integración con Otros Sistemas:

- Inventario: Los ítems encontrados se guardan
- Combate: Los enemigos inician peleas
- Experiencia: Los eventos dan XP
- Economia: Los tesoros dan monedas
- Quirks: Los quirks se añaden al personaje

COMENZAR TRABAJO

ORDEN RECOMENDADO:

- 1. Estudiar el código base proporcionado
- 2. Modificar sistema de países (categorías)
- 3. Integrar con estructura userData existente
- 4. Conectar con sistemas de inventario/combate
- 5. Crear comandos slash necesarios
- 6. Testear funcionalidad completa

ARCHIVOS A PRIORIZAR:

- 1. (time-weather.js) Modificar selección de países
- 2. (exploration.js) Adaptar recompensas
- 3. (character-creation.js) Añadir categorías de países
- 4. Bot principal Añadir event handlers

¡El código base está sólido y bien estructurado! Tu trabajo es principalmente de integración y adaptación al bot existente.